

```
/* modo.h

Copyright (c) 1993-2012. Free Software Foundation, Inc.

This file is part of GNU MCSim.

GNU MCSim is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 3
of the License, or (at your option) any later version.

GNU MCSim is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with GNU MCSim; if not, see <http://www.gnu.org/licenses/>

-- Revisions -----
Logfile: %F%
Revision: %I%
Date: %G%
Modtime: %U%
Author: @a
-- SCCS -----

Header file for outputting routines.
*/
#ifndef MODO_H_DEFINED

/* -----
-----
Constants */

#define ALL_VARS (0)

/* -----
-----
Typedefs */

typedef int (*PFI_CALLBACK) (PFILE, PVMMAPSTRCT, PVOID);

/* -----
-----
Macros */

#define WriteIndexName(pfile, pvm) (fprintf ((pfile), "ID_%s", (pvm)->szName))
```

```

/* -----
Prototypes */

int AdjustOneVar (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
void AdjustVarHandles (PVMMAPSTRCT pvmGlo);
int AssertExistsEqn (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
int CountOneDecl (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
int ForAllVar (PFILE pfile, PVMMAPSTRCT pvm, PFI_CALLBACK pfiFunc,
               HANDLE hType, PVOID pinfo);
int ForAllVarwSep (PFILE pfile, PVMMAPSTRCT pvm, PFI_CALLBACK pfiFunc,
                   HANDLE htype, PVOID pinfo);
PSTR GetName (PVMMAPSTRCT pvm, PSTR szModelVarName, PSTR szDerivName,
              HANDLE hType);
int IndexOneVar (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
void IndexVariables (PVMMAPSTRCT pvmGlo);
void ReversePointers (PVMMAPSTRCT *ppvm);
void TranslateEquation (PFILE pfile, PSTR szEqn, long iEqType);
void TranslateID (PINPUTBUF pibDum, PFILE pfile, PSTR szLex, int
iEqType);
void VerifyEqns (PVMMAPSTRCT pvmGlo, PVMMAPSTRCT pvmDyn);
void WriteCalcDeriv (PFILE pfile, PVMMAPSTRCT pvmGlo, PVMMAPSTRCT
pvmDyn);
void WriteCalcJacob (PFILE pfile, PVMMAPSTRCT pvmGlo, PVMMAPSTRCT
pvmJacob);
void WriteCalcOutputs (PFILE pfile, PVMMAPSTRCT pvmGlo,
                      PVMMAPSTRCT pvmCalcOut);
void WriteDecls (PFILE pfile, PVMMAPSTRCT pvmGlo);
void WriteHeader (PFILE pfile, PSTR szName, PVMMAPSTRCT pvmGlo);
void WriteIncludes (PFILE pfile);
void WriteInitModel (PFILE pfile, PVMMAPSTRCT pvmGlo);
void WriteModel (PINPUTINFO pinfo, PSTR szFileOut);
int WriteOneDecl (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
int WriteOneEquation (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
int WriteOneIndexDefine (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
int WriteOneInit (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
int WriteOneName (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
int WriteOneOutputName (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
int WriteOneVMEEntry (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
int WriteOne_R_PIDefine (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
int WriteOne_R_SODefine (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
int WriteOne_R_ParmInit (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
int WriteOne_R_PSDecl (PFILE pfile, PVMMAPSTRCT pvm, PVOID pInfo);
void WriteScale (PFILE pfile, PVMMAPSTRCT pvmGlo, PVMMAPSTRCT pvmScale);
void WriteVarMap (PFILE pfile, PVMMAPSTRCT pvmGlo);
void Write_R_CalcDeriv (PFILE pfile, PVMMAPSTRCT pvmGlo, PVMMAPSTRCT
pvmDyn,
                       PVMMAPSTRCT pvmCalcOut);
void Write_R_CalcJacob (PFILE pfile, PVMMAPSTRCT pvmGlo,
                       PVMMAPSTRCT pvmJacob);
void Write_R_Decls (PFILE pfile, PVMMAPSTRCT pvmGlo);
void Write_R_Events (PFILE pfile, PVMMAPSTRCT pvmGlo, PVMMAPSTRCT
pvmEvents);

```

```
void Write_R_Includes (PFILE pfile);
void Write_R_InitModel (PFILE pfile, PVMMAPSTRCT pvmGlo);
void Write_R_InitPOS (PFILE pfile, PVMMAPSTRCT pvmGlo, PVMMAPSTRCT
pvmScale);
void Write_R_Model (PINPUTINFO pinfo, PSTR szFileOut);
void Write_R_Roots (PFILE pfile, PVMMAPSTRCT pvmGlo, PVMMAPSTRCT
pvmRoots);
void Write_R_Scale (PFILE pfile, PVMMAPSTRCT pvmGlo, PVMMAPSTRCT
pvmScale);
void Write_R_State_Scale (PFILE pfile, PVMMAPSTRCT pvmScale);

#define MODO_H_DEFINED
#endif

/* End */
```